Balanced-by-construction regular and ω-regular languages

Luc Edixhoven[0000−0002−6011−9535] (✉) and Sung-Shik Jongmans[0000−0002−4394−8745]

[1] Open University, Heerlen, Netherlands
[2] Centrum Wiskunde & Informatica (CWI), Amsterdam, Netherlands
{led,ssj}@ou.nl

Abstract. $\mathrm{Paren_n}$ is the typical generalisation of the Dyck language to multiple types of parentheses. We generalise its notion of balancedness to allow parentheses of different types to freely commute. We show that balanced regular and ω-regular languages can be characterised by syntactic constraints on regular and ω-regular expressions and, using the shuffle on trajectories operator, we define grammars for balanced-by-construction expressions with which one can express every balanced regular and ω-regular language.

Keywords: Dyck language · Shuffle on trajectories · Regular languages

1 Introduction

The Dyck language of balanced parentheses is a textbook example of a context-free language. Its typical generalisation to multiple types of parentheses, $\mathrm{Paren_n}$, is central in characterising the class of context-free languages, as shown by the Chomsky-Schützenberger theorem [1]. Many other generalisations of the Dyck language have been studied over the years [2,4,5,8,9].

The notion of balancedness in $\mathrm{Paren_n}$ requires parentheses of different types to be properly nested: $[_1 [_2]_2]_1$ is balanced but $[_1 [_2]_1]_2$ is not. In this paper, we consider a more general notion of balancedness, in which parentheses of the same type must be properly nested but parentheses of different types may freely commute. This notion of balancedness is of particular interest in the context of distributed computing, where different components communicate by exchanging messages: if we assign a unique type of parentheses to every communication channel between two participants, and interpret a left parenthesis as a message send event and a right parenthesis as a receive event, then balancedness characterises precisely all sequences of communication with no lost or orphan messages.

Specifically, we are interested in specifying languages that are balanced by construction, which correspond to communication protocols that are free of lost and orphan messages. More precisely, we aim to answer the question: can we define balanced atoms and a set of balancedness-preserving operators with which one can express all balanced languages?

Our main result is that we answer this question positively for the classes of regular and ω-regular languages. Our contributions are as follows:

- In Section 2 we show how balancedness of regular languages corresponds to syntactic properties of regular expressions.
- In Section 3 we show that, by using a parametrised shuffle operator, we can define a grammar of balanced-by-construction expressions with which one can express all balanced regular languages.
- In Section 4 we extend these results to ω-regular languages and expressions.

Related work and detailed proofs appear in a technical report [3].

Notation $\mathbb{N} = \{1, 2, \ldots\}$, $\mathbb{N}_0 = \{0, 1, \ldots\}$ and $\mathbb{Z}$ is the set of integers. Let Σ_n be the alphabet $\{[_1,]_1, \ldots, [_n,]_n\}$. Its size is typically clear from the context, in which case we omit the subscript. We write λ for the empty word. We write Σ^* for the set of finite words over Σ. We write Σ^ω for the set of infinite words $\{w \mid w : \mathbb{N} \to \Sigma\}$ over Σ. We write $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. We write $w(i)$ to refer to the symbol at position i in w. We write $w(i, \ldots, j)$ for the substring of w beginning at position i and ending at position j. Let $v, w \in \Sigma^\infty$. Then v is a prefix of w, denoted $v \preceq w$, if $v = w$ or if there exists $v' \in \Sigma^\infty$ such that $vv' = w$. We write $|w|, |w|_\sigma \in \mathbb{N}_0 \cup \{\aleph_0\}$ respectively for the length of w and for the number of occurrences of symbol σ in w. Let $\mathbb{E}$ be the set of all regular expressions over $\bigcup_{n \geq 1} \Sigma_n$. For $e_1, e_2 \in \mathbb{E}$, we write $e_1 \equiv e_2$ iff $L(e_1) = L(e_2)$.

2 Balanced regular languages

In this section, we formally define our notion of balancedness and characterise balanced regular languages in terms of regular expressions.

Balancedness A word $w \in \Sigma^*$ is *i-balanced* if $|w|_{[_i} = |w|_{]_i}$ and if, for all prefixes v of w, $|v|_{[_i} \geq |v|_{]_i}$. It is *balanced* if it is i-balanced for all i. We extend this terminology to languages and expressions in the expected way.

Regular expressions Using standard algebraic rules, we can rewrite any regular expression representing a non-empty language into an equivalent expression that does not contain $\varnothing$. Therefore, without loss of generality, we may assume that regular expressions do not contain $\varnothing$, unless they are simply $\varnothing$.

To every regular expression e and for every i, we assign a value which we call its *i-balance*, denoted $\nabla(e, i)$. We show that this value corresponds to the number of unmatched left i-parentheses in every word of its language (see Lemma 1(i)), if such a number exists. Also, to differentiate between words such as $[_i]_i$ and $]_i[_i$, we assign a second value to regular expressions which we call its *minimum i-balance*, denoted $\nabla^{\min}(e, i)$, which we show to correspond to the smallest i-balance among every prefix of every word in its language (see Lemma 1(ii–iii)).

Formally, we define partial functions $\nabla, \nabla^{\min} : \mathbb{E} \times \mathbb{N} \mapsto \mathbb{Z}$ as in Figure 1. Lemma 1 states that $\nabla(e, i)$ and $\nabla^{\min}(e, i)$ have the intended properties we

$$\nabla(\lambda, i) = 0 \qquad \nabla(\texttt{[}_i, i) = 1 \qquad \nabla(\texttt{]}_i, i) = -1 \qquad \nabla(\texttt{[}_j, i) = \nabla(\texttt{]}_j, i) = 0$$
$$\nabla(e_1 + e_2, i) = \nabla(e_1, i) \text{ if } \nabla(e_1, i) = \nabla(e_2, i)$$
$$\nabla(e_1 \cdot e_2, i) = \nabla(e_1, i) + \nabla(e_2, i) \qquad \nabla(e^*, i) = 0 \text{ if } \nabla(e, i) = 0$$
$$\nabla^{\min}(\lambda, i) = \nabla^{\min}(\texttt{[}_i, i) = 0 \qquad \nabla^{\min}(\texttt{]}_i, i) = -1 \qquad \nabla^{\min}(\texttt{[}_j, i) = \nabla^{\min}(\texttt{]}_j, i) = 0$$
$$\nabla^{\min}(e_1 + e_2, i) = \min(\nabla^{\min}(e_1, i), \nabla^{\min}(e_2, i))$$
$$\nabla^{\min}(e_1 \cdot e_2, i) = \min(\nabla^{\min}(e_1, i), \nabla(e_1, i) + \nabla^{\min}(e_2, i)) \qquad \nabla^{\min}(e^*, i) = \nabla^{\min}(e, i)$$

Fig. 1. The i-balance and minimum i-balance of regular expressions, where $i \neq j$.

described and Lemma 2 states that if the number of unmatched i-parentheses of words in $L(e)$ is uniquely defined, then both $\nabla(e, i)$ and $\nabla^{\min}(e, i)$ are defined. We note that ∇ is partial. For instance, $\nabla(\texttt{[}_1 + \lambda, 1)$ and $\nabla(\texttt{[}_1^*, 1)$ are both undefined since their languages contain $\texttt{[}_1$ and λ, which have different numbers of unmatched left i-parentheses. As $\nabla^{\min}$ relies on ∇, $\nabla^{\min}$ is partial as well.

Lemma 1. *If $\nabla(e, i)$ and $\nabla^{\min}(e, i)$ are defined, then:*

(i) $|w|_{\texttt{[}_i} - |w|_{\texttt{]}_i} = \nabla(e, i)$ *for every $w \in L(e)$;*
(ii) $|v|_{\texttt{[}_i} - |v|_{\texttt{]}_i} \geq \nabla^{\min}(e, i)$ *for every prefix v of every $w \in L(e)$; and*
(iii) $|v|_{\texttt{[}_i} - |v|_{\texttt{]}_i} = \nabla^{\min}(e, i)$ *for some prefix v of some $w \in L(e)$.*

Lemma 2. *If $|v|_{\texttt{[}_i} - |v|_{\texttt{]}_i} = |w|_{\texttt{[}_i} - |w|_{\texttt{]}_i}$ for every $v, w \in L(e)$ and $L(e) \neq \varnothing$, then $\nabla(e, i)$ and $\nabla^{\min}(e, i)$ are defined.*

The proofs are straightforward by structural induction on e. Applying them gives us the following characterisation:

Theorem 1. *Let $e \in \mathbb{E}$. Then e is balanced iff $\nabla(e, i) = \nabla^{\min}(e, i) = 0$ for every i or if $e = \varnothing$.*

3 Balanced-by-construction regular languages

The main contribution of this section is a grammar of balanced-by-construction expressions, $\mathbb{E}^{\sqcup\!\sqcup}$ in Figure 2, with which one can express all balanced regular languages. It differs from regular expressions in two ways:

- Parentheses can syntactically occur only in ordered pairs instead of separately, so the atoms are all balanced.
- We add a family of operators $\sqcup\!\sqcup_\theta^n(e_1, \ldots, e_n)$, called *shuffle on trajectories*, in order to interleave words of subexpressions.

The shuffle on trajectories operator is a powerful variation of the traditional shuffle operator, which adds a control trajectory (or a set thereof) to restrict the permitted orders of interleaving. This allows for fine-grained control over orderings when shuffling words or languages. The binary operator was defined

$$e ::= \varnothing \mid \lambda \mid [_1 \cdot]_1 \mid [_2 \cdot]_2 \mid \ldots \mid e_1 + e_2 \mid e_1 \cdot e_2 \mid e^* \mid \shuffle^1_\theta(e_1) \mid \shuffle^2_\theta(e_1, e_2) \mid \ldots$$
$$\theta ::= \varnothing \mid \lambda \mid 1 \mid 2 \mid \ldots \mid \theta_1 + \theta_2 \mid \theta_1 \cdot \theta_1 \mid \theta^*$$

Fig. 2. A grammar $\mathbb{E}^{\shuffle}$ for expressing balanced regular languages.

— and its properties thoroughly studied — by Mateescu et al. [6]; the slightly later introduced multiary variant [7] is formally defined as follows.

Let $w_1, \ldots, w_n \in \Sigma^*$ and let $t \in \{1, \ldots, n\}^*$ be a *trajectory*. Then:

$$\shuffle^n_t(w_1, \ldots, w_n) = \begin{cases} \sigma \shuffle^n_{t'}(w_1, \ldots, w'_i, \ldots, w_n) & \text{if } t = it' \wedge w_i = \sigma w'_i, \\ \lambda & \text{if } t = w_1 = \ldots = w_n = \lambda. \end{cases}$$

The operator naturally generalises to languages and expressions:

$$\shuffle^n_T(L_1, \ldots, L_n) = \{\shuffle^n_t(w_1, \ldots, w_n) \mid t \in T, w_1 \in L_1, \ldots, w_n \in L_n\}.$$
$$L(\shuffle^n_\theta(e_1, \ldots, e_n)) = \shuffle^n_{L(\theta)}(L(e_1), \ldots, L(e_n)).$$

As the operator's arity is clear from its operands, we generally omit it. For the trajectories, we allow any regular expression over $\mathbb{N}$.

Note that $\shuffle^n_t(w_1, \ldots, w_n)$ is defined only if $|t|_i = |w_i|$ for every i. If $|t|_i = |w_i|$, we say that t *fits* w_i. For example, $\shuffle_{121332}([_1]_1, [_2]_2, [_3]_3) = [_1[_2]_1[_3]_3]_2$ and $\shuffle_{121}([_1]_1, [_2]_2)$ is undefined since 121 does not fit $[_2]_2$. Similarly, $\shuffle_{12+21}([_1 + [_2]_2,]_1) \equiv [_1]_1 +]_1[_1$, $\shuffle_{12+22}([_1,]_1) \equiv [_1]_1$ and $\shuffle_{(12)^*}(([_1]_1)^*, ([_2]_2)^*) \equiv ([_1[_2]_1]_2)^*$, while $\shuffle_{12+11}([_1, \lambda) \equiv \shuffle_{(12)^*}([_1]_1, [_2(]_2[_2)^*) \equiv \varnothing$ since in both cases no trajectory fits at least one word in every operand. Additionally, we say that T fits L_i if *every* $t \in T$ fits some $w_i \in L_i$ and that θ fits e_i if $L(\theta)$ fits $L(e_i)$.

In the remainder of this section, we show that the grammar $\mathbb{E}^{\shuffle}$ can express all (completeness) and only (soundness) balanced regular languages.

Soundness Showing that every expression in $\mathbb{E}^{\shuffle}$ represents a balanced regular language is straightforward. The base cases all comply and both balanced and regular languages are closed under nondeterministic choice, concatenation and finite repetition. The shuffle on trajectories operator yields an interleaving of its operands: a simple inductive proof will show closure of balanced languages under the operation. Mateescu et al. show that regular languages are closed under binary shuffle on regular trajectory languages by constructing an equivalent finite automaton [6, Theorem 5.1]; their construction can be generalised in a straightforward way to fit the multiary operator, which shows that:

Theorem 2. *$\{L(e) \mid e \in \mathbb{E}^{\shuffle}\} \subseteq \{L \mid L$ is a balanced and regular language$\}$.*

Completeness To show that every balanced regular language has a representation in $\mathbb{E}^{\shuffle}$, we take a balanced regular expression e, rewrite it into a disjunctive normal form $e_1 + \ldots + e_n$ such that all e_i contain no $\varnothing$ or choice operators — unless $e = \varnothing$, but since $\varnothing \in \mathbb{E}^{\shuffle}$ we do not need to consider that specific case.

$$\langle\,\rangle_i^k = (\mathtt{[}_i\mathtt{]}_i)^k(\mathtt{[}_i\mathtt{]}_i)^* \qquad \langle\lambda\rangle_i^k = (\langle\,\rangle_i^k)^* \qquad \langle\,\rangle_i^\omega = (\mathtt{[}_i\mathtt{]}_i)^\omega$$

$$\langle+\rangle_i^k = \langle\,\rangle_i^k\mathtt{[}_i \qquad \langle-\rangle_i^k = \mathtt{]}_i\langle\,\rangle_i^k \qquad \langle\pm\rangle_i^k = \mathtt{]}_i\langle\,\rangle_i^k\mathtt{[}_i \qquad \langle\star\rangle_i^k = (\langle\pm\rangle_i^k)^* \qquad \langle\pm\rangle_i^\omega = (\mathtt{]}_i\mathtt{[}_i)^\omega$$

Fig. 3. Factors, with $i \in \mathbb{N}, k \in \mathbb{N}_0$; balanced factors in the top row, unbalanced factors in the bottom row. We omit the superscript when it is not relevant. The ω-factors will be used in Section 4.

$$(\langle+\rangle_i^k, \langle-\rangle_i^\ell) \to \langle\,\rangle_i^{k+\ell+1} \qquad (\langle+\rangle_i^k, \langle\pm\rangle_i^\ell) \to \langle+\rangle_i^{k+\ell+1} \qquad (\langle\pm\rangle_i^k, \langle-\rangle_i^\ell) \to \langle-\rangle_i^{k+\ell+1}$$

$$(\langle-\rangle_i^k, \langle+\rangle_i^\ell) \to \langle\pm\rangle_i^{k+\ell} \qquad (\langle+\rangle_i^k, \langle\star\rangle_i^\ell) \to \langle+\rangle_i^{k} \qquad (\langle\star\rangle_i^k, \langle-\rangle_i^\ell) \to \langle-\rangle_i^{\ell}$$

$$(\langle\pm\rangle_i^k, \langle\pm\rangle_i^\ell) \to \langle\pm\rangle_i^{k+\ell+1} \qquad (\langle\star\rangle_i^k, \langle\star\rangle_i^\ell) \to \langle\star\rangle_i^{\min(k,\ell)} \qquad (\langle\pm\rangle_i^k, \langle\star\rangle_i^\ell), (\langle\star\rangle_i^\ell, \langle\pm\rangle_i^k) \to \langle\pm\rangle_i^{k}$$

$$(\langle+\rangle_i^k, \langle\pm\rangle_i^\omega) \to \langle\,\rangle_i^\omega$$

Fig. 4. Merging common pairs of factors, with $i \in \mathbb{N}$ and $k, \ell \in \mathbb{N}_0$.

We then show that, for every i, $e_i \equiv ⧢_\theta(e_{i,1}, \ldots, e_{i,m})$ for some $e_{i,1}, \ldots, e_{i,m}$, where every $e_{i,j}$ is essentially of the form $(\mathtt{[}_k\mathtt{]}_k)^*$ for some k.

To do this, we show the more general result that, in fact, any regular expression containing no $\varnothing$ or $+$, and whose every i-balance is defined, can be written as the shuffle of the expressions in Figure 3, which we call *factors*. Additionally, this can be done in such a way that the number of unbalanced i-factors is limited by the expression's i-balance and minimum i-balance, which implies that if the expression is balanced then it can be written as a shuffle of balanced factors — which is in $\mathbb{E}^{⧢}$. To prove this inductively for the concatenation case, we use that $⧢_{\theta_1}(e_1, \ldots, e_n) \cdot ⧢_{\theta_2}(e_{n+1}, \ldots, e_{n+m}) \equiv ⧢_{\theta_3}(e_1, \ldots, e_n, e_{n+1}, \ldots, e_{n+m})$ for some θ_3. We then merge certain pairs of factors to retain the correspondence between unbalanced factors and i-balance; for example, $\langle+\rangle_i$ and $\langle-\rangle_i$ into $\langle\,\rangle_i$.

Lemma 3 justifies this merging operation and specifies the conditions under which it may be applied. We note that in particular these conditions, with the right T, hold for the pairs of factors in Figure 4. Using this, Lemma 4 justifies the rewriting of regular expressions into shuffles of factors.

Lemma 3 (Merge). *Let $L = ⧢_T(L_1, \ldots, L_m)$. If*

(a) *T fits every L_i,*
(b) *for every $t \in T$, if $t(i) = m-1$ and $t(j) = m$ then $i < j$, and*
(c) *for all $v, w \in L_{m-1}L_m$, if $|v| = |w|$ then $v = w$,*

then $L = ⧢_{T'}(L_1, \ldots, L_{m-1}L_m)$ for some T' such that T' fits $L_1, \ldots, L_{m-1}L_m$.

Proof. Let φ be a homomorphism such that $\varphi(m-1) = 1$, $\varphi(m) = 2$ and $\varphi(i) = \lambda$ for all other i. Let ψ be a homomorphism such that $\psi(m) = m-1$ and $\psi(i) = i$ for all other i. We proceed to show that $L = ⧢_{\psi(T)}(L_1, \ldots, L_{m-1}L_m)$. Since T fits every L_i, $\psi(T)$ also fits $L_1, \ldots, L_{m-1}L_m$. $\square$

Lemma 4 (Rewrite). *Let $\mathsf{pos}_i(e_1, \ldots, e_n)$, $\mathsf{neg}_i(e_1, \ldots, e_n)$, $\mathsf{neut}_i(e_1, \ldots, e_n)$ be the number of $\langle+\rangle_i$, $\langle-\rangle_i$ and $[\langle\pm\rangle_i$ or $\langle\star\rangle_i]$ among $e_1, \ldots, e_n$.*

Let $e \in \mathbb{E}$ containing no $+$, whose i-balance is defined for every i. Then there exist θ and factors $e_1, \dots, e_n$ such that $e \equiv \sqcup\!\sqcup_\theta(e_1, \dots, e_n)$ and, additionally,

(a) $\mathsf{pos}_i(e_1, \dots, e_n) - \mathsf{neg}_i(e_1, \dots, e_n) = \nabla(e, i)$ *for every* i,
(b) $-\mathsf{neg}_i(e_1, \dots, e_n) - \mathsf{neut}_i(e_1, \dots, e_n) = \nabla^{\min}(e, i)$ *for every* i,
(c) *there are not both* $\langle + \rangle_i$ *and* $\langle - \rangle_i$ *among* $e_1, \dots, e_n$ *for some* i, *and*
(d) θ *fits every* e_i.

Proof. This is a proof by induction on the structure of e.

The base cases λ, $\texttt{[}_i$ and $\texttt{]}_i$ are covered by $\sqcup\!\sqcup^1_\lambda(\langle\,\rangle^0_i)$, $\sqcup\!\sqcup^1_1(\langle + \rangle^0_i)$ and $\sqcup\!\sqcup^1_1(\langle - \rangle^0_i)$. Since e contains no $+$, this leaves us with two inductive cases:

- Let $e = \hat{e}^*$. The induction hypothesis gives us some $\hat{e}_1, \dots, \hat{e}_n$ and $\hat{\theta}$ satisfying all conditions for $\hat{e}$. It should be clear that $L((\sqcup\!\sqcup_{\hat{\theta}}(\hat{e}_1, \dots, \hat{e}_n))^*) \subseteq L((\sqcup\!\sqcup_{\hat{\theta}}(\hat{e}^*_1, \dots, \hat{e}^*_n))^*) \subseteq L(\sqcup\!\sqcup_{\hat{\theta}^*}(\hat{e}^*_1, \dots, \hat{e}^*_n))$. Since $\nabla(e, i)$ is defined for all i, $\nabla(\hat{e}, i) = 0$ for all i. It then follows from (a) and (c) that $\hat{e}_1, \dots, \hat{e}_n$ contain no $\langle + \rangle_i$ or $\langle - \rangle_i$, so all $\hat{e}^*_i$ are also factors.
  To prove inclusion in the other direction, we show in two steps that $L(\sqcup\!\sqcup_{\hat{\theta}^*}(\hat{e}^*_1, \dots, \hat{e}^*_n)) \subseteq L((\sqcup\!\sqcup_{\hat{\theta}}(\hat{e}^*_1, \dots, \hat{e}^*_n))^*) \subseteq L((\sqcup\!\sqcup_{\hat{\theta}}(\hat{e}_1, \dots, \hat{e}_n))^*)$.
  The balances, minimum balances and factor counts are unchanged, so (a–c) are satisfied. Finally, since $\hat{\theta}$ fits every $\hat{e}_i$, $\hat{\theta}^*$ fits every $\hat{e}^*_i$, so (d) also holds.
- Let $e = \hat{e}_1 \cdot \hat{e}_2$. The induction hypothesis gives us some $e_{1,1}, \dots, e_{1,m_1}$ and θ_1 satisfying all conditions for $\hat{e}_1$, and similarly for $\hat{e}_2$. Let φ be a homomorphism such that $\varphi(i) = i + m_1$. Then $e' = \sqcup\!\sqcup_{\theta_1\varphi(\theta_2)}(e_{1,1}, \dots, e_{1,m_1}, e_{2,1}, \dots, e_{2,m_2}) \equiv e$ and e' satisfies (d), but not necessarily (a–c). We resolve the latter by merging operands $e_{1,j}, e_{2,k}$ where applicable by Lemma 3. We merge pairs of factors from Figure 4, taking care to prioritise pairs containing both $\langle + \rangle_i$ and $\langle - \rangle_i$ over pairs containing only one of these, and pairs containing only one over pairs containing none. By Lemma 3, the resulting expression is equivalent to e' and satisfies (d). It also satisfies (a–c). $\square$

Since a balanced regular expression has an i-balance and minimum i-balance of 0 for every i (Theorem 1), the following theorem follows directly from Lemma 4.

Theorem 3. $\{L(e) \mid e \in \mathbb{E}^{\sqcup\!\sqcup}\} \supseteq \{L \mid L$ *is a balanced and regular language*$\}$.

As an example, consider $e = \texttt{[}_1(\texttt{[}_1\texttt{]}_1 + \texttt{]}_1\texttt{[}_1)(\texttt{]}_1\texttt{[}_1)^*\texttt{]}_1$. We first rewrite e as $\texttt{[}_1\texttt{[}_1\texttt{]}_1(\texttt{]}_1\texttt{[}_1)^*\texttt{]}_1 + \texttt{[}_1\texttt{]}_1\texttt{[}_1(\texttt{]}_1\texttt{[}_1)^*\texttt{]}_1$. We proceed to show how to construct an expression in $\mathbb{E}^{\sqcup\!\sqcup}$ for the first part of the disjunction:

$$\begin{aligned}
\texttt{[}_1\texttt{[}_1\texttt{]}_1(\texttt{]}_1\texttt{[}_1)^*\texttt{]}_1 &\equiv \underline{\sqcup\!\sqcup_1(\langle + \rangle^0_1)\ \sqcup\!\sqcup_1\ (\langle + \rangle^0_1)}\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)(\sqcup\!\sqcup_1(\langle - \rangle^0_1)\ \sqcup\!\sqcup_1\ (\langle + \rangle^0_1))^*\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)\\
&\equiv \underline{\sqcup\!\sqcup_{12}(\langle + \rangle^0_1, \langle + \rangle^0_1)\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)}(\sqcup\!\sqcup_1(\langle - \rangle^0_1)\ \sqcup\!\sqcup_1\ (\langle + \rangle^0_1))^*\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)\\
&\equiv \sqcup\!\sqcup_{121}(\langle\,\rangle^1_1, \langle + \rangle^0_1)(\underline{\sqcup\!\sqcup_1(\langle - \rangle^0_1)\ \sqcup\!\sqcup_1\ (\langle + \rangle^0_1)})^*\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)\\
&\equiv \sqcup\!\sqcup_{121}(\langle\,\rangle^1_1, \langle + \rangle^0_1)\underline{(\sqcup\!\sqcup_{11}(\langle \pm \rangle^0_1))^*}\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)\\
&\equiv \underline{\sqcup\!\sqcup_{121}(\langle\,\rangle^1_1, \langle + \rangle^0_1)\ \sqcup\!\sqcup_{(11)^*}\ (\langle \star \rangle^0_1)}\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)\\
&\equiv \underline{\sqcup\!\sqcup_{121(22)^*}(\langle\,\rangle^1_1, \langle + \rangle^0_1)\ \sqcup\!\sqcup_1\ (\langle - \rangle^0_1)}\\
&\equiv \sqcup\!\sqcup_{121(22)^*2}(\langle\,\rangle^1_1, \langle\,\rangle^1_1).
\end{aligned}$$

4 Balanced-by-construction ω-regular languages

We generalise the notion of balancedness to also include bounded infinite words and ω-languages: a word $w \in \Sigma^\infty$ is *i-balanced* iff $|w|_{\texttt{[}_i} = |w|_{\texttt{]}_i}$, $|v|_{\texttt{[}_i} \geq |v|_{\texttt{]}_i}$ for all *finite* prefixes v of w, and w is bounded, as defined below. A language $L \subseteq \Sigma^\infty$ is i-balanced if all of its words are and if it is bounded. This is extended to balancedness and expressions in the expected way. We note that all finite words and balanced regular languages are bounded by default; boundedness is only a restriction on infinite words and ω-languages.[3]

Boundedness A word $w \in \Sigma^\infty$ is *i-bounded by $n \in \mathbb{N}_0$* if $|v|_{\texttt{[}_i} - |v|_{\texttt{]}_i} \leq n$ for all finite prefixes v of w. A language is i-bounded by n if all of its words are. A word or language is *bounded* if it is i-bounded for all i. The *minimal i-bound* of a word or language is the smallest n for which it is i-bounded. We extend these definitions to expressions in the expected way.

We note that by this definition $\texttt{[}_i(\texttt{[}_i\texttt{]}_i)^\omega$ is balanced, but $\texttt{[}_i^*(\texttt{[}_i\texttt{]}_i)^\omega$ is not since it is not bounded, even though all of its words are.

4.1 Balanced ω-regular expressions

We use Ω for the set of all ω-regular expressions. It is defined as follows:

$$\frac{}{\varnothing \in \Omega} \qquad \frac{e \in \mathbb{E} \quad \lambda \notin L(e)}{e^\omega \in \Omega} \qquad \frac{e_1 \in \mathbb{E} \quad e_2 \in \Omega}{e_1 \cdot e_2 \in \Omega} \qquad \frac{e_1, e_2 \in \Omega}{e_1 + e_2 \in \Omega} \tag{1}$$

As before, we assume without loss of generality that an ω-regular expression e does not contain $\varnothing$, unless $e = \varnothing$, to simplify definitions and proofs.

Our characterisation of balanced ω-regular expressions is a generalisation of that of balanced regular expressions. We note two main complications:

- We need to distinguish between finite and infinite numbers of parentheses: $\texttt{[}_1(\texttt{[}_1\texttt{]}_1)^\omega$ is balanced but $\texttt{[}_1(\texttt{[}_2\texttt{]}_2)^\omega$ is not. We introduce two predicates for expressions: $\xi(e,i)$ and $\xi^\omega(e,i)$, as defined in Figure 5. Intuitively, and as shown in Lemma 5, $\xi(e,i)$ iff every word in $L(e)$ contains at least one i-parenthesis, and $\xi^\omega(e,i)$ iff every word in $L(e)$ contains infinitely many.
- Not every subexpression of a balanced ω-regular expression can be assigned a unique i-balance: $(\lambda + \texttt{[}_i)(\texttt{[}_i\texttt{]}_i)^\omega$ is balanced, but $(\lambda + \texttt{[}_i)$ has no unique i-balance. Instead, we now assign an upper bound ∇^U and a lower bound ∇^L to an expression's i-balance instead of a single value. These are defined in Figure 6. The definition of minimum i-balance is unchanged, other than the addition of $\nabla^{\min}(e^\omega, i) = \nabla^{\min}(e,i)$ and the redefinition of $\nabla^{\min}(e_1 \cdot e_2, i) = \min(\nabla^{\min}(e_1,i), \nabla^L(e_1,i) + \nabla^{\min}(e_2,i))$. We note that, for any regular expression $e \in \mathbb{E}$, $\nabla^L(e,i) = \nabla^U(e,i) = \nabla(e,i)$.

[3] Our choice for boundedness stems from our interest in communication protocols (Section 1), where channels often require buffers of finite size.

$$\frac{}{\xi([_i, i)} \qquad \frac{}{\xi(]_i, i)} \qquad \frac{\xi(e_1, i) \vee \xi(e_2, i)}{\xi(e_1 \cdot e_2, i)} \qquad \frac{\xi(e_1, i) \quad \xi(e_2, i)}{\xi(e_1 + e_2, i)} \qquad \frac{\xi(e, i)}{\xi(e^\omega, i)}$$

$$\frac{\xi^\omega(e_2, i)}{\xi^\omega(e_1 \cdot e_2, i)} \qquad \frac{\xi^\omega(e_1, i) \quad \xi^\omega(e_2, i)}{\xi^\omega(e_1 + e_2, i)} \qquad \frac{\xi(e, i)}{\xi^\omega(e^\omega, i)}$$

Fig. 5. The i-occurrence of regular and ω-regular expressions.

$$\nabla^\dagger(\lambda, i) = 0 \qquad \nabla^\dagger([_i, i) = 1 \qquad \nabla^\dagger(]_i, i) = -1 \qquad \nabla^\dagger([_j, i) = \nabla^\dagger(]_j, i) = 0$$

$$\nabla^\dagger(e_1 \cdot e_2, i) = \begin{cases} \nabla^\dagger(e_2, i) & \text{if } \xi^\omega(e_2, i) \\ \nabla^\dagger(e_1, i) + \nabla^\dagger(e_2, i) & \text{otherwise} \end{cases}$$

$$\nabla^\dagger(e^*, i) = \nabla^\dagger(e^\omega, i) = 0 \text{ if } \nabla^\dagger(e, i) = 0$$

$$\nabla^L(e_1 + e_2, i) = \min(\nabla^L(e_1, i), \nabla^L(e_2, i)) \quad \nabla^U(e_1 + e_2, i) = \max(\nabla^U(e_1, i), \nabla^L(e_2, i))$$

Fig. 6. The i-balance bounds of ω-regular expressions, where $i \neq j$ and $\dagger \in \{L, U\}$.

Lemma 5. *Let $e \in \mathbb{E} \cup \Omega$ such that $e \neq \varnothing$. Then:*

(i) *$\xi(e, i)$ if and only if $|w|_{[_i} + |w|_{]_i} > 0$ for every $w \in L(e)$;*
(ii) *$\xi^\omega(e, i)$ if and only if $|w|_{[_i} + |w|_{]_i} = \aleph_0$ for every $w \in L(e)$.*

We extend Lemmas 1 and 2 about properties of i-balance and minimum i-balance to i-balance bounds and ω-regular expressions in Lemmas 6 and 7.

Lemma 6 (cf. Lemma 1). *Let $e \in \mathbb{E} \cup \Omega$. If $\nabla^L(e, i)$, $\nabla^U(e, i)$ and $\nabla^{\min}(e, i)$ are defined, then:*

(i) *For every $w \in L(e)$, $|w|_{[_i}$ and $|w|_{]_i}$ are either both finite or both infinite;*
(ii) *For every $w \in L(e)$, if $|w|_{[_i}, |w|_{]_i}$ are finite, then $\nabla^L(e, i) \leq |w|_{[_i} - |w|_{]_i} \leq \nabla^U(e, i)$;*
(iii) *If $e \in \mathbb{E}$, then there exist $w_1, w_2 \in L(e)$ such that $|w_1|_{[_i} - |w_1|_{]_i} = \nabla^L(e, i)$ and $|w_2|_{[_i} - |w_2|_{]_i} = \nabla^U(e, i)$;*
(iv) *If $\xi^\omega(e, i)$, then $\nabla^L(e, i) = \nabla^U(e, i) = 0$;*
(v) *$|v|_{[_i} - |v|_{]_i} \geq \nabla^{\min}(e, i)$ for every finite prefix v of every $w \in L(e)$;*
(vi) *$|v|_{[_i} - |v|_{]_i} = \nabla^{\min}(e, i)$ for some finite prefix v of some $w \in L(e)$;*
(vii) *$L(e)$ is i-bounded.*

Lemma 7 (cf. Lemma 2). *Let $e \in \mathbb{E} \cup \Omega$. If $e \neq \varnothing$, e is i-bounded and if there exists some n such that $|(|v|_{[_i} - |v|_{]_i}) - (|w|_{[_i} - |w|_{]_i})| \leq n$ for all $v, w \in L(e)$ with finite i-parenthesis counts, then $\nabla^L(e, i)$, $\nabla^U(e, i)$ and $\nabla^{\min}(e, i)$ are defined.*

The proofs are straightforward by structural induction on e. Applying these lemmas gives us the following characterisation:

Theorem 4. *Let $e \in \mathbb{E} \cup \Omega$. Then e is balanced iff $\nabla^L(e, i) = \nabla^U(e, i) = \nabla^{\min}(e, i) = 0$ for every i or if $e = \varnothing$.*

4.2 Balanced-by-construction ω-regular languages

The grammar in Figure 2 can be extended with ω as in (1) to obtain an expression grammar $\Omega^{\sqcup\!\sqcup}$ for balanced ω-regular languages [3, Appendix B].

Since the inductive definition of the shuffle on trajectories operator does not support words of infinite length, we redefine it as follows. Let $w_1, \ldots, w_n \in \Sigma^\infty$ and let $t \in \{1, \ldots, n\}^\infty$. If t fits $w_1, \ldots, w_n$, i.e., if $|t|_i = |w_i|$ for every i, then $\sqcup\!\sqcup_t(w_1, \ldots, w_n) = w(1)w(2)\ldots w(|t|)$ if t has finite length and $w(1)w(2)\ldots$ if t has infinite length, where $w(i) = w_j(k)$ for $j = t(i)$ and $k = |t(1, \ldots, i)|_j$. As before, this naturally extends to languages and expressions.

Soundness Balanced languages being closed under shuffle follows immediately from its definition. To show that $\sqcup\!\sqcup_T(L_1, \ldots, L_n)$ is ω-regular if T is ω-regular and all L_i are either regular or ω-regular, we can further generalise the construction used by Mateescu et al. [6] to build a Muller automaton for the resulting language. Recall that a Muller automaton differs from a finite automaton only in its acceptance criterion: instead of a single set of final states it has a set of sets of final states F, and it accepts all infinite words for which the set of states that are visited infinitely often is an element of F.

The construction of the new Muller automaton is analogous to the construction of a finite automaton for a shuffle of regular languages and differs only in the construction of F. Let Q be the set of states of our new Muller automaton. Let F_i be the acceptance criterion of the automaton for L_i, whether a finite automaton or a Muller automaton. If L_i is regular, then without loss of generality we may assume that no state in F_i has any outgoing transition. Furthermore, since ω-regular languages are closed under intersection and the language of all trajectories containing infinitely many i is ω-regular for every i, we may also assume without loss of generality that T only contains trajectories with infinitely many occurrences of every i.

We define F as the cross product of all the F_i: F is the set of sets of states such that, if L_i is ω-regular then the projection of these states on i is an element of F_i, and if L_i is regular then the projection of these states on i is a single state in F_i. Formally: if $\varphi_i((q_t, q_1, \ldots, q_n)) = q_i$ and $\varphi_i(S) = \{\varphi_i(q) \mid q \in S\}$, then $F = \{S \mid S \subseteq Q \wedge (\varphi_i(S) \in F_i \vee (\varphi_i(S) \subseteq F_i \wedge |\varphi_i(S)| = 1))\}$. The automaton for T forces that every Muller automaton for some L_i takes infinitely many steps. By our assumption that the final states of finite automata have no outgoing transitions, all finite automata only take a finite number of steps. It follows that our constructed Muller automaton accepts the language of $\sqcup\!\sqcup_T(L_1, \ldots, L_n)$, which then is ω-regular. In other words:

Theorem 5. *$\{L(e) \mid e \in \Omega^{\sqcup\!\sqcup}\} \subseteq \{L \mid L$ is a balanced ω-regular language$\}$.*

Completeness Our approach to showing that every balanced ω-regular expression has an equivalent expression in $\Omega^{\sqcup\!\sqcup}$ mirrors that of Section 3: we first rewrite an expression into a disjunctive normal form and then recursively construct an expression in $\Omega^{\sqcup\!\sqcup}$ for every term of the disjunction by merging pairs of factors.

Let $e \neq \varnothing$ be a balanced ω-regular expression. Without loss of generality, we may assume that $e = e_1 e_2^\omega + \ldots + e_{2m-1} e_{2m}^\omega$, where every e_i is a regular expression containing no $+$. Otherwise, we can rewrite it as such. We show how to construct an expression in $\Omega^{\sqcup\!\sqcup}$ for $e_1 e_2^\omega$.

Since $\nabla^L(e,i) = \nabla^U(e,i) = \nabla^{\min}(e,i) = 0$ by Theorem 4, it follows that $\nabla^{\min}(e_1,i) = \nabla^L(e_2,i) = \nabla^U(e_2,i) = 0$. Then, by Lemma 4, we can write e_1 as a shuffle of $\langle\,\rangle_i, \langle\lambda\rangle_i, \langle+\rangle_i$ and e_2 as a shuffle of $\langle\,\rangle_i, \langle\lambda\rangle_i, \langle\pm\rangle_i, \langle\star\rangle_i$. The idea is to: (a) rewrite e_2^ω in terms of $\langle\,\rangle_i, \langle\lambda\rangle_i, \langle\,\rangle_i^\omega, \langle\pm\rangle_i^\omega$ and then; (b) merge every $\langle+\rangle_i$ in e_1 with a $\langle\pm\rangle_i^\omega$ in e_2^ω into $\langle\,\rangle_i^\omega$, using Lemma 3. We run into two complications:

- In step (a), e_2^ω may not necessarily be expressible as a single shuffle of factors: if $e_2 = {[}_1{]}_1({[}_2{]}_2)^*$, then e_2^ω contains both words with finite and infinite numbers of ${[}_2, {]}_2$. The latter requires a factor $\langle\,\rangle_2^\omega$, while the former requires its absence. To remedy this, we write e_2^ω as a *disjunction* of shuffles of factors; one for every combination of finite and infinite versions of $\langle\,\rangle_i, \langle\lambda\rangle_i$. This is further detailed in Lemma 8.
- In step (b), the number of $\langle\pm\rangle_i^\omega$ in a term of e_2^ω may not necessarily match the number of $\langle+\rangle_i$ in e_1: if $e_1 = {[}_1$ and $e_2 = {[}_1{]}_1$, then e_1 contains one $\langle+\rangle_1$ and e_2 contains one factor $\langle\,\rangle_1$. To solve this, we use two observations:
  - We can apply Lemma 3 to split a $\langle\,\rangle_i$ into $\langle+\rangle_i$ and $\langle-\rangle_i$.
  - $e_2^\omega \equiv (e_2 \cdot e_2)^\omega$, so we can essentially multiply the factors in e_2.

  Thus, we can always split a $\langle\,\rangle_i$ into $\langle+\rangle_i$ and $\langle-\rangle_i$, then create copies of them and merge them back into one $\langle\,\rangle_i$ and one $\langle\pm\rangle_i$. Since we can merge all other factors with their own copy, this effectively adds one $\langle\pm\rangle_i$. Now that we have at least one, we can create more: we create a copy of every factor, then merge every factor with its own copy except for some number of $\langle\pm\rangle_i$. This is further detailed in Lemma 9.

Lemma 8. *Let $e = \sqcup\!\sqcup_\theta(e_1,\ldots,e_n) \in \mathbb{E}^{\sqcup\!\sqcup}$ be a shuffle of factors $\langle\,\rangle_i, \langle\lambda\rangle_i, \langle\pm\rangle_i$ such that θ fits every e_j and contains no $+$. Then $e^\omega \equiv \hat{e}_1 + \ldots + \hat{e}_m$, where $\hat{e}_k = \sqcup\!\sqcup_{\theta_k}(e_{k,1},\ldots,e_{k,n})$ is a shuffle of factors $\langle\,\rangle_i, \langle\lambda\rangle_i, \langle\,\rangle_i^\omega, \langle\pm\rangle_i^\omega$ for every k such that the number of $\langle\pm\rangle_i$ in e is the same as the number of $\langle\pm\rangle_i^\omega$ in $\hat{e}_k$ for every i, and θ_k fits every $e_{k,j}$.*

Proof. Let $\varphi : \mathbb{E} \mapsto 2^{\mathbb{E}\cup\Omega}$ such that $\varphi(\langle\,\rangle_i^k) = \{\langle\,\rangle_i^k, \langle\,\rangle_i^\omega\}$, $\varphi(\langle\lambda\rangle_i^k) = \{\langle\lambda\rangle_i^k, \langle\,\rangle_i^\omega\}$ and $\varphi(\langle\pm\rangle_i^k) = \{\langle\pm\rangle_i^\omega\}$. We can then show that $e^\omega \equiv \hat{e}_1 + \ldots + \hat{e}_m$, where $\{\hat{e}_1,\ldots,\hat{e}_m\} = \{\sqcup\!\sqcup_{\theta^\omega}(e'_1,\ldots,e'_n) \mid e'_1 \in \varphi(e_1),\ldots,e'_n \in \varphi(e_n)\}$.

Moreover, since φ maps $\langle\pm\rangle_i$ to $\langle\pm\rangle_i^\omega$, the number of factors $\langle\pm\rangle_i^\omega$ in every $\hat{e}_k$ matches the number of factors $\langle\pm\rangle_i$ in e. However, if $\hat{e}_k = \sqcup\!\sqcup_{\theta^\omega}(e'_1,\ldots,e'_n)$, then θ^ω may not necessarily fit every e'_j: if e'_j is one of $\langle\,\rangle_i, \langle\lambda\rangle_i$, then there are $t \in L(\theta^\omega)$ with infinitely many j, while every word in $L(e'_j)$ is finite. Instead of θ^ω, we can use the trajectory $\theta^* \cdot \psi(\theta)^\omega$, where ψ is a homomorphism such that $\psi(j) = \lambda$ if e'_j is one of $\langle\,\rangle_i, \langle\lambda\rangle_i$ and $\psi(j) = j$ otherwise. This covers exactly the part of θ^ω that fits every e'_j. □

Lemma 9. *Let $\sqcup\!\sqcup_\theta(e_1,\ldots,e_n) \equiv e \in \mathbb{E}$ be a shuffle of factors $\langle\,\rangle_i, \langle\lambda\rangle_i, \langle\pm\rangle_i, \langle\star\rangle_i$ such that θ fits every e_j and contains no $+$, and $\xi(e,i)$. If there are ℓ factors*

$\langle\pm\rangle_i, \langle\star\rangle_i$ *among* $e_1, \ldots, e_n$*, then for every* $k \geq \ell$ *(such that* $k > 0$*), there exists some shuffle of factors* $\hat{e} = \sqcup\!\sqcup_{\hat{\theta}}(\hat{e}_1, \ldots, \hat{e}_m)$ *such that* $e^\omega \equiv \hat{e}^\omega$*,* $\hat{e}$ *contains* k *factors* $\langle\pm\rangle_i$ *and no* $\langle\star\rangle_i$ *and* $\hat{\theta}$ *fits every* $\hat{e}_j$*.*

Proof. This proof consists of three steps. First, we need to make sure that we have at least one $\langle\pm\rangle_i$. Second, we replace any remaining factors $\langle\star\rangle_i$ with $\langle\pm\rangle_i$. Third, we create additional copies of $\langle\pm\rangle_i$ as needed.

1. Suppose that there are no $\langle\pm\rangle_i$ among $e_1, \ldots, e_n$. Then our first step consists of creating one. Since $\xi(e, i)$ and θ contains no $+$, there exists some $e_j \in \{\langle\,\rangle_i, \langle\lambda\rangle_i, \langle\star\rangle_i\}$ such that $|t|_j > 0$ for every $t \in L(\theta)$. Without loss of generality, we may assume that $j = n$.
   If $e_n = \langle\star\rangle_i^k$, since $|t|_n > 0$ for every t then $e \equiv \sqcup\!\sqcup_\theta(e_1, \ldots, \langle\pm\rangle_i^k)$ and we can proceed with step 2. Otherwise, if $e_n = \langle\lambda\rangle_i^k$, then $e \equiv \sqcup\!\sqcup_\theta(e_1, \ldots, \langle\,\rangle_i^k)$ and if $e_n = \langle\,\rangle_i^0$, then $e \equiv \sqcup\!\sqcup_\theta(e_1, \ldots, \langle\,\rangle_i^1)$. Going forward, we may thus assume that $e_n = \langle\,\rangle_i^k$ with $k \geq 1$. Since $|t|_n > 0$ for every $t \in L(\theta)$ and θ contains no $+$, it follows that $\theta = \theta_1 \cdot \theta_2$ such that both θ_1 and θ_2 only contain trajectories with odd numbers of n. We can then apply the proof of Lemma 3 to show that $e \equiv \sqcup\!\sqcup_{\theta_3}(e_1, \ldots, e_{n-1}, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2})$ for some θ_3, k_1, k_2. If $e_1, \ldots, e_{n-1}$ contain a $\langle\star\rangle_i$, then without loss of generality we may assume that $e_{n-1} = \langle\star\rangle_i^{k_3}$. We may assume that there exists some $t \in L(\theta)$ such that $|\theta|_{n-1} = 0$; otherwise we would have selected this factor as e_n earlier in this step and then proceeded with step 2. It follows that all trajectories in θ_1 and θ_2, and therefore in θ_3, contain even numbers of n. Then, in the same way that we split $\langle\,\rangle_i^k$ into $\langle+\rangle_i^{k_1}$ and $\langle-\rangle_i^{k_2}$ before, we can show that $e \equiv \sqcup\!\sqcup_{\theta_4}(e_1, \ldots, e_{n-2}, \langle\star\rangle_i^{k_4}, \langle\star\rangle_i^{k_5}, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2})$ for some θ_4, k_4, k_5. As seen in Figure 4, we can then merge $\langle\star\rangle_i^{k_4}$ with $\langle-\rangle_i^{k_2}$ and $\langle\star\rangle_i^{k_5}$ with $\langle+\rangle_i^{k_1}$ to obtain $e \equiv \sqcup\!\sqcup_{\theta_5}(e_1, \ldots, e_{n-2}, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2})$ for some θ_5. This takes care of the special case where $k = \ell > 0$ but there are no factors $\langle\pm\rangle_i$. We may thus assume without loss of generality that $e \equiv \sqcup\!\sqcup_{\theta_6}(e_1, \ldots, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2})$ for some θ_6.
   Since we still lack a $\langle\pm\rangle_i$, we use that $e^\omega \equiv (e \cdot e)^\omega$ to construct $e' = \sqcup\!\sqcup_{\theta_6}(e_1, \ldots, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2}) \cdot \sqcup\!\sqcup_{\theta_6}(e_1, \ldots, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2}) \equiv \sqcup\!\sqcup_{\theta_7}(e_1, \ldots, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2}, e_1, \ldots, \langle+\rangle_i^{k_1}, \langle-\rangle_i^{k_2})$ for some θ_7. We can then merge the first $\langle+\rangle_i^{k_1}$ with the second $\langle-\rangle_i^{k_2}$ into $\langle\,\rangle_i^{k_1+k_2+1}$ and merge the second $\langle+\rangle_i^{k_1}$ with the first $\langle-\rangle_i^{k_2}$ into $\langle\pm\rangle_i^{k_1+k_2}$. We can merge every other factor with its own copy, which gives us $e' \equiv \sqcup\!\sqcup_{\theta_8}(e'_1, \ldots, \langle\,\rangle_i^{k_1+k_2+1}, \langle\pm\rangle_i^{k_1+k_2})$ and $e_1'^\omega \equiv e^\omega$.
2. Now that we have at least one $\langle\pm\rangle_i$, we can reuse methods applied in the first step to replace any remaining $\langle\star\rangle_i$: create a copy of every factor using $e^\omega \equiv (e \cdot e)^\omega$, then merge the two copies of $\langle\star\rangle_i$ with the copies of some $\langle\pm\rangle_i$ as in Figure 4. By merging every other factor with its own copy, we effectively replace one $\langle\star\rangle_i$ with one $\langle\pm\rangle_i$. We repeat this step until there are no $\langle\star\rangle_i$ left.
3. Finally, by copying every factor and then merging every factor with its own copy except for a number of $\langle\pm\rangle_i$, we can create any additional number of $\langle\pm\rangle_i$, until we have some $\hat{e} = \sqcup\!\sqcup_{\hat{\theta}}(\hat{e}_1, \ldots, \hat{e}_m)$ with k $\langle\pm\rangle_i$. Since every rewriting step preserves equivalence of the ω-closures and the fitting of the trajectories, it follows that $\hat{e}^\omega \equiv e^\omega$ and that $\hat{\theta}$ fits every $\hat{e}_j$. $\square$

Summarising, given $e_1 \cdot e_2^\omega$, by applying Lemmas 9 and 8 we can rewrite e_1 as a shuffle of factors $\langle\,\rangle_i, \langle \lambda \rangle_i, \langle + \rangle_i$, and e_2^ω as a disjunction of shuffles of factors $\langle\,\rangle_i, \langle \lambda \rangle_i, \langle\,\rangle_i^\omega, \langle \pm \rangle_i^\omega$, such that the number of $\langle \pm \rangle_i^\omega$ in every term of the disjunction equals the number of $\langle + \rangle_i$ in e_1. By applying the laws of distributivity, we can then rewrite $e_1 \cdot e_2^\omega$ as a disjunction of concatenations of shuffles. Since the numbers of $\langle + \rangle_i$ and $\langle \pm \rangle_i^\omega$ match in every term of this disjunction, we can apply Lemma 3 to merge every pair into $\langle\,\rangle_i^\omega$. Since all factors are now balanced, every balanced ω-regular language has a corresponding expression in $\Omega^{ш}$:

Theorem 6. $\{L(e) \mid e \in \Omega^{ш}\} \supseteq \{L \mid L$ *is a balanced* ω*-regular language*$\}$.

As an example, we show how to build an expression in $\Omega^{ш}$ for $e = \texttt{[}_1(\texttt{[}_1\texttt{]}_1)^\omega$.

$$\begin{aligned}
\texttt{[}_1(\texttt{[}_1\texttt{]}_1)^\omega &\equiv ш_1(\langle + \rangle_1^0)(\underline{ш_{11}(\langle\,\rangle_1^1)})^\omega \\
&\equiv ш_1(\langle + \rangle_1^0)\underline{(ш_1(\langle + \rangle_1^0)\ ш_1\ (\langle - \rangle_1^0))^\omega} \\
&\equiv ш_1(\langle + \rangle_1^0)(ш_1(\langle + \rangle_1^0)\underline{ш_1(\langle - \rangle_1^0)\ ш_1\ (\langle + \rangle_1^0)}\ ш_1\ (\langle - \rangle_1^0))^\omega \\
&\equiv ш_1(\langle + \rangle_1^0)(\underline{ш_1(\langle + \rangle_1^0)\ ш_{11}\ (\langle \pm \rangle_1^0)\ ш_1\ (\langle - \rangle_1^0)})^\omega \\
&\equiv ш_1(\langle + \rangle_1^0)\underline{(ш_{1221}(\langle\,\rangle_1^1, \langle \pm \rangle_1^0))^\omega} \\
&\equiv \underline{ш_1(\langle + \rangle_1^0)\ ш_{(1221)^\omega}\ (\langle\,\rangle_1^\omega, \langle \pm \rangle_1^\omega)} \\
&\equiv ш_{1(2112)^\omega}(\langle\,\rangle_1^\omega, \langle\,\rangle_1^\omega).
\end{aligned}$$

References

1. Chomsky, N., Schützenberger, M.: The algebraic theory of context-free languages. In: Computer Programming and Formal Systems, Studies in Logic and the Foundations of Mathematics, vol. 26, pp. 118 – 161. Elsevier (1959)
2. Duchon, P.: On the enumeration and generation of generalized Dyck words. Discret. Math. **225**(1-3), 121–135 (2000). https://doi.org/10.1016/S0012-365X(00)00150-3
3. Edixhoven, L., Jongmans, S.S.: Balanced-by-construction regular and ω-regular languages (technical report). Tech. Rep. OUNL-CS-2021-1, Open University of the Netherlands (2021)
4. Labelle, J., Yeh, Y.: Generalized Dyck paths. Discret. Math. **82**(1), 1–6 (1990). https://doi.org/10.1016/0012-365X(90)90039-K
5. Liebehenschel, J.: Lexicographical generation of a generalized Dyck language. SIAM J. Comput. **32**(4), 880–903 (2003). https://doi.org/10.1137/S0097539701394493
6. Mateescu, A., Rozenberg, G., Salomaa, A.: Shuffle on trajectories: Syntactic constraints. Theor. Comput. Sci. **197**(1-2), 1–56 (1998). https://doi.org/10.1016/S0304-3975(97)00163-1
7. Mateescu, A., Salomaa, K., Yu, S.: On fairness of many-dimensional trajectories. J. Autom. Lang. Comb. **5**(2), 145–157 (2000). https://doi.org/10.25596/jalc-2000-145
8. Moortgat, M.: A note on multidimensional Dyck languages. In: Categories and Types in Logic, Language, and Physics. Lecture Notes in Computer Science, vol. 8222, pp. 279–296. Springer (2014). https://doi.org/10.1007/978-3-642-54789-8_16
9. Prodinger, H.: On a generalization of the Dyck-language over a two letter alphabet. Discret. Math. **28**(3), 269–276 (1979). https://doi.org/10.1016/0012-365X(79)90134-1